

## **TMOD PLANNING & SEQUENCING SUBSYSTEM USERS GUIDE**

**May 12, 2000**

**Revision: Preliminary - Software Delivery E1.0**

**Prepared by:**

**Steven Wissler & Judy Morris  
TMOD Mission Sequencing Systems Engineering**



**Jet Propulsion Laboratory  
California Institute Of Technology  
4800 Oak Grove Drive  
Pasadena, CA 91109-8099**

<b>1. INTRODUCTION.....</b>	<b>1-1</b>
1.1 TTACS Function.....	1-1
1.2 SOPC Function .....	1-1
1.3 Sequence Function Allocations .....	1-1
1.3.1 ASP Initiation.....	1-1
1.3.2 Sequence Generation.....	1-2
1.3.3 Sequence Translation .....	1-2
1.3.4 RSOE/DKF Generation.....	1-2
1.4 Applicable Documents.....	1-3
1.4.1 TMOD Software User's Guides .....	1-3
1.4.2 Software Interface Specification Modules .....	1-3
<b>2. SEQUENCE ENVIRONMENT AND USER SETUP .....</b>	<b>2-1</b>
2.1 Environment Variables .....	2-1
2.2 File Naming Conventions .....	2-2
2.3 Software Configuration File Locations.....	2-2
2.4 Sequence PDB Input Data File Locations.....	2-3
2.5 Project Specific Sequence Information.....	2-3
2.5.1 Processor Id's.....	2-3
2.5.2 Seqtran Directives .....	2-5
2.6 NIPC and EC SASF Formats .....	2-8
2.7 MGS - Special Payload Blocks.....	2-8
2.8 MGS - Special Spacecraft Blocks .....	2-9
<b>3. SEQUENCE SOFTWARE FUNCTIONS.....</b>	<b>3-12</b>
3.1 Core Software .....	3-12
3.1.1 seqgen .....	3-12
3.1.2 seqtran - MGS Only .....	3-13
3.1.3 sts - (slink).....	3-13
3.1.4 apgen.....	3-13
3.1.5 seq_review .....	3-14

<b>3.2      Automation Tools .....</b>	<b>3-15</b>
3.2.1     run.apgen.....	3-15
3.2.2     check_sASF .....	3-16
3.2.3     wrap_sASF .....	3-17
3.2.4     send_nipcec .....	3-18
3.2.5     send_ic .....	3-19
3.2.6     send_frf .....	3-20
3.2.7     gen_command .....	3-21
3.2.8     env_gen .....	3-23
3.2.9     seq_translate.....	3-24
3.2.10    file_load .....	3-25
3.2.11    printSCMF .....	3-26
3.2.12    revinit_gen .....	3-27
3.2.13    run_revtran .....	3-28
3.2.14    revtran .....	3-29
3.2.15    revcompare.....	3-31
3.2.16    run_revcompare .....	3-32
3.2.17    get_pef_errors .....	3-33
3.2.18    pef_display .....	3-34
3.2.19    pdb_store.....	3-36
3.2.20    notify_act .....	3-37
3.2.21    gen_dsn_config_codes .....	3-38
3.2.22    run_dsn_config.....	3-39
3.2.23    get_seqtran_info.....	3-40
3.2.24    run_sOE .....	3-41
3.2.25    gen_sOE_dkf .....	3-42
3.2.26    correct_transmitter_min_dur .....	3-43
3.2.27    finish_sOE_dkf.....	3-44
3.2.28    notify_sOE .....	3-45
3.2.29    ftp_dsn.....	3-46
3.2.30    ftp_2homepage.....	3-47
<b>4.      SEQUENCE DAEMON FUNCTIONS .....</b>	<b>4-1</b>
<b>4.1     Sequence Daemon .....</b>	<b>4-1</b>
<b>4.2     auto_seq_status .....</b>	<b>4-1</b>
4.2.1     Display Screen .....	4-1
4.2.2     Menu Items .....	4-1
<b>4.3     start_auto_sequence .....</b>	<b>4-3</b>
<b>4.4     stop_auto_sequence .....</b>	<b>4-4</b>
<b>5.      APPENDIX A .....</b>	<b>5-1</b>

## **1. Introduction**

The TMOD Planning and Sequencing Subsystem (PSS) is distributed over 3 functional areas, Sequencing Team (SEQ/SCT), Test Telemetry and Command System (TTACS) and Science Operations Planning Computer/Experiment Rep (SOPC/ER ).

### **1.1 TTACS Function**

The TTACS workstations will be used to generate, store and transmit real-time commands and sequences for S/C testing. The command products may be generated ahead of time and re-called for transmission, or may be generated and transmitted on the fly.

### **1.2 SOPC Function**

The SOPC/ER workstations will be used for generating command requests (SASFs) and sending them to the Automated Sequence Processor (ASP) system for processing. Stored sequence SASFs will be transferred to the Project Database (PDB) by the PI and/or ER and the PST will be notified via E-Mail or phone. Non-Interactive Payload Commands (NIPC) SASFs will be transferred to the PDB and the ASP will be notified via an E-mail File Release Form (EFRF). The SASFs may be created by non-PSS software. SEQGEN is provided to the SOPC/ER workstations and can be used to check and/or edit SASFs created by non-PSS software, or can be used to create SASF files from scratch.

### **1.3 Sequence Function Allocations**

#### **1.3.1 ASP Initiation**

<b>Function</b>	<b>SEQ Team</b>	<b>SOPC/ER</b>	<b>S/C Team</b>
<b>check_sasf</b>	x	x	x
<b>wrap_sasf</b>	x	x	x
<b>send_nipcec</b>	x	x	x
<b>send_ic</b>	x	x	x
<b>send_frf</b>	x	x	x

### 1.3.2 Sequence Generation

Function	SEQ Team	SOPC/ER	S/C Team
<b>gen_command</b>	x	x	x
<b>env_gen</b>	x	x	x
<b>seqgen</b>	x	x	x

### 1.3.3 Sequence Translation

Function	SEQ Team	SOPC/ER	S/C Team
<b>seq_translate</b>	x		x

### 1.3.4 RSOE/DKF Generation

Function	SEQ Team	SOPC/ER	S/C Team
<b>pef_display</b>	x		
<b>run_soe</b>	x		
<b>gen_soe_dkf</b>	x		
<b>correct_transmitter_min_dur</b>	x		
<b>finish_soe_dkf</b>	x		
<b>notify_soe</b>	x		
<b>ftp_dsn</b>	x		
<b>ftp_2homepage</b>	x		

## **1.4 Applicable Documents**

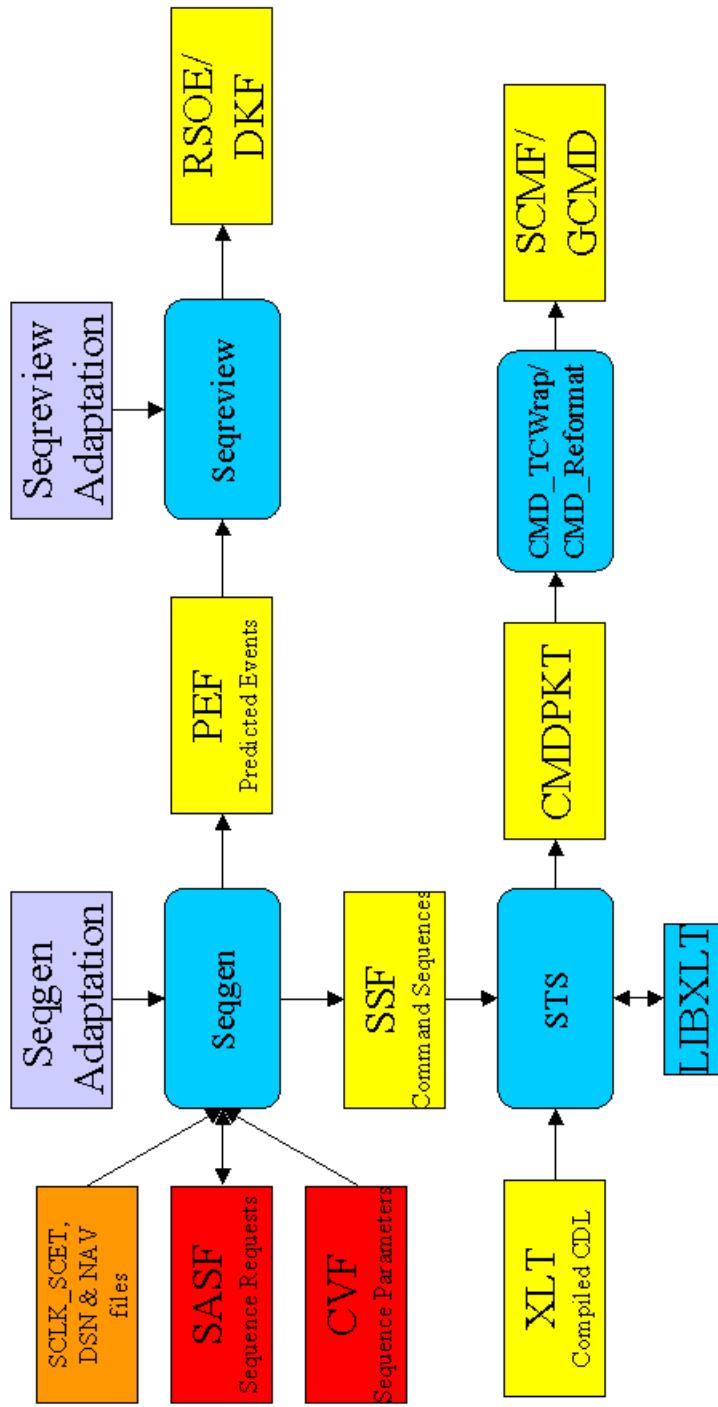
### **1.4.1 TMOD Software User's Guides**

TSEQ0729-01-00-21	SEQGEN/State-Tracker V23 Users Guide	<a href="http://seq-www/docs/SE/SST/SSTug.html">http://seq-www/docs/SE/SST/SSTug.html</a>
MSEQ0720-02-00-17	SEQTRAN V22 Users's Guide	<a href="http://div390-www/ammos_docs/docs/v22/seq_ug/seqtran.pdf">http://div390-www/ammos_docs/docs/v22/seq_ug/seqtran.pdf</a>
MSEQ0723-01-00-18	APGEN V1.0 User's Guide	<a href="http://div390-www/ammos_docs/docs/v22/seq_ug/seqtran.pdf">http://div390-www/ammos_docs/docs/v22/seq_ug/seqtran.pdf</a>
TSEQ0729-01-00-17	SEQ REVIEW V4.0 User's Guide	<a href="http://div390-www/ammos_docs/docs/v23/seq_ug/seq_rev.pdf">http://div390-www/ammos_docs/docs/v23/seq_ug/seq_rev.pdf</a>

### **1.4.2 Software Interface Specification Modules**

SFOC-1-SEQ-Any-SASF	Spacecraft Activity Sequence File	<a href="http://div390-www/ammos_docs/docs/v22.4/sis/seq/seqanysf.pdf">http://div390-www/ammos_docs/docs/v22.4/sis/seq/seqanysf.pdf</a>
SFOC-3-SEQ-Environ	Environment File	<a href="http://div390-www/ammos_docs/docs/v22.4/pgnt/environ.pdf">http://div390-www/ammos_docs/docs/v22.4/pgnt/environ.pdf</a>
SFOC-3-SEQ-CVDF	Context Variable Definition File	<a href="http://div390-www/ammos_docs/docs/v22/pgnt/cvdf.pdf">http://div390-www/ammos_docs/docs/v22/pgnt/cvdf.pdf</a>
SFOC-3-SEQ-Legend	Legend File	
SFOC-3-SEQ-Conditions	Conditions File	<a href="http://div390-www/ammos_docs/docs/v23.1/pgnt/condtns.pdf">http://div390-www/ammos_docs/docs/v23.1/pgnt/condtns.pdf</a>

## TMOD Core components and File Interfaces



## 2. Sequence Environment and User Setup

The sequence scripts and programs use UNIX environment variables to determine the location and names of files. The file which defines these environment variables is **msop-seq-setup**, or **sirtf-seq-setup**. This file can be referenced in the .cshrc file as:

### For MSOP users:

```
source /msop/seq/bin/msop-seq-setup
```

### For SIRTF users:

```
source /afs/jpl.nasa.gov/group/sirtf/stfcm/flt/uplink/seq/bin/sirtf-seq-setup
```

### For FST users:

```
source /afs/jpl.nasa.gov/group/tmod_upl_adapt/seq/bin/tmod-seq-setup
```

### 2.1 Environment Variables

This lists the key environment variables set up by the setup script.

Environment Variable	Default Value (MSOP)	Function
DaemonNode	mgseq2	W/S running Sequence Daemon
DaemonUser	nipcec	Username for Sequence Daemon
DSN_DIR	/pub/nss/soe	Subdirectory to send DKF's
DSN_NODE	dsn	DSN server's node name
HOME_PAGE_DIR	/local/etc/httpd/htdocs/seq	Subdirectory to send SOE's
HOME_PAGE_NODE	mgsyw3	Seq team's homepage node name
HOME_PAGE_SIE	<a href="http://mgsyw3.jpl.nasa.gov/seq">http://mgsyw3.jpl.nasa.gov/seq</a>	URL address of Seq team's homepage
LD_LIBRARY_PATH	/msop/seq/bin: /sfoc/lib	Unix run-time library search path
ORT_TEST	FALSE	Set to TRUE for ORT tests. Sends _SIM s/c DKF's to dsn server, and uses regular mail list.
PSS_BASE_PATH	/msop/seq	Base path for CM configured sequence software
PSS_DATA_PATH	/msop/seq_data	Base path for User modifiable sequence data
SEQ_EXE_DIR	/msop/seq/bin	Directory containing sequence software and scripts
XUSERFILESEARCHPATH	/msop/seq/bin/%N: /local/app-defaults/%N	X resource file search path

## 2.2 File Naming Conventions

All sequence automation scripts assume a certain file naming convention. Different sequence products have specific file “extensions”. The extensions are all lower case. The scripts assume these extensions and it is not necessary to specify the extensions when running the scripts.

Type	Extension	Description
<b>SASF</b>	.sASF	Spacecraft Activity Sequence File
<b>ENV</b>	.env	Seq Environment File
<b>PEF</b>	.pef	Predicted Events File
<b>SSF</b>	.ssf	Spacecraft Sequence File
<b>FINCON</b>	.fincon	Final Conditions File
<b>RSOE</b>	.rsoe	Real-Time Sequence of Events
<b>DKF</b>	.dkf	DSN Keyword File
<b>SCMF</b>	.scmf	Spacecraft Message File
<b>DMWF</b>	.dmwf	Desired Memory Word File
<b>SYMOUT</b>	.symout	SEQTRAN symbols File
<b>CMDPKT</b>	.cmdpkt	Command Packet File
<b>RPEF</b>	.rpef	Reverse Translated PEF

## 2.3 Software Configuration File Locations

The local versions of these configuration files are located within the \$PSS\_CONFIG\_PATH directory tree.

Directory	Used By	Description
<a href="#">sc_tables</a> slink_config dsn_config lander_power_model seq_review_scripts seqgen_scripts distiller engineering tools	get_sc_info slink seqgen seqgen & apgen pef_display seqgen finish_soe	Contains spacecraft dependant configuration data. Configuration files for STS. Configuration files used for DSN modeling. Configuration files for the Mars 98 Lander power model  Contains engineering versions of delivered scripts. Contains scripts which are not under CM control.

## **2.4 Sequence PDB Input Data File Locations**

The local versions of these sequence input files are located within the \$PSS\_INPUT\_PATH directory tree. These files should be pulled from the PDB upon receipt and stored in these locations. The function env\_gen will automatically determine which input files are necessary for a sequence, and if they do not exist in the local directories, it will pull them off of the PDB and store them in the proper locations. The following table lists the locations.

Directory	File Description
sclk	<b>Spacecraft Clock Coefficients File</b>
ltf	<b>Light Time File</b>
optg	<b>Orbit Propagation and Timing Geometry File</b>
vue	<b>DSN Viewperiods File</b>
alloc	<b>DSN Allocations File</b>

## **2.5 Project Specific Sequence Information**

### **2.5.1 Processor Id's**

Processor Id's are used by seqtran to direct the uplink information flow to the spacecraft.

#### **2.5.1.1 MGS**

Proc	Function
SSC	<b>Stored Sequence Commands</b>
RTC	<b>Real-Time Commands, both processors</b>
PRI	<b>Real-Time Commands, primary processor only</b>
SEC	<b>Real-Time Commands, secondary processor only</b>
CON	<b>For Contingency Mode Enabled Sequences</b>

#### 2.5.1.2 Mars 01 Orbiter

Proc	Function
VC0	<b>Hardware Immediate Commands</b>
VC1	<b>Software Immediate Commands</b>
VC2	<b>File Loads</b>
VC5	<b>End of Session Indicator</b>
VC6	<b>MSP 01 Lander Direct Data Pass-Through Command Message</b>

#### 2.5.1.3 Stardust

Proc	Function
VC0	<b>Hardware Immediate Commands</b>
VC1	<b>Software Immediate Commands</b>
VC2	<b>File Loads</b>
VC5	<b>End of Session Indicator</b>

#### 2.5.1.4 Genesis

Proc	Function
VC0	<b>Hardware Immediate Commands</b>
VC1	<b>Software Immediate Commands</b>
VC2	<b>File Loads</b>
VC5	<b>End of Session Indicator</b>

## 2.5.2 Seqtran Directives

### 2.5.2.1 MGS

#### WINDOW

The WINDOW directive tells seqtran the sequence type and the uplink window opportunities. It is required for stored sequences and mini-sequences.

Parameter Name	Description	Req./Opt.
OPEN	Uplink window open time for this sequence	Required
CLOSE	Uplink window close time for this sequence	Required
OPEN_P1	Next uplink window open time	Required
OPEN_P2	Uplink window open time following OPEN_P1	Required
WIND_TYPE	NORMAL -Normal Stored Sequence INHIB - Sequence is not automatically started by SEQEND. DRAG - Drag Pass Sequence MINSEQ - Mini Sequence RESET - Sequence Memory is reset to an empty sequence area using current buffer and script counts. RELOAD - Mission dependant scripts are reloaded prior to loading the sequence REINIT - RESET and RELOAD are performed	Required

#### SEQEND

Required for any sequence which uses the WINDOW directive. SEQEND is responsible for completing the memory management process and constructing a complete sequence to be sent to the spacecraft. It is not required for Real-time commands.

No Parameters.

## MSEQST

The MSEQST directive tells seqtran the special mini-sequence type and load address for load-only sequences.

Parameter Name	Description	Req./Opt.
SEQTYP	LOAD - Load only at address LOADGO - Load and start in dedicated memory SAFE - Create Safe mode mini-sequence	Required
ADDRESS	Start address of type LOAD	Required for LOAD

## MSEQND

The MSEQND must be paired with the MSEQST directive and must come after the last command in the mini-sequence.

No Parameters.

## SETTER

Parameter Name	Description	Req./Opt.
LABEL		Required
OPCODE		Required
PARAMS		Required

### 2.5.2.2 Mars 01, Stardust & Genesis

## SEQSTR

The SEQSTR directive tells seqtran to build a sequence file for the commands in the SASF/SSF.

Parameter Name	Description	Req./Opt.
BEGIN_TIME	Uplink window open	Required
END_TIME	Uplink window close	Required
TIME_TYPE	“ABSLTE” - Absolute time sequences “RELATV” - Relative time sequences	Required
SEQ_NAME	Sequence File Name on S/C	Required
PATH	File Path on S/C	Required

## **BRANCH**

The BRANCH directive is a user-level interface the S/C command SEQ\_BRANCH. It allows input of a mnemonic label, which sequence translation will convert into a relative offset and invoke the SEQ\_BRANCH command.

<b>Parameter Name</b>	<b>Description</b>	<b>Req./Opt.</b>
CONDITION	Sequence Branch Condition “UNCOND” “NEVER” “EQ” “NE” “LT” “LE” “GT” “GE” “BAND” “BOR” “BXOR” “LAND” “LOR” “LXOR”	Required
SEQ_VAR_TYPE1	Sequence Variable #1 Type “GLOBAL” or “LOCAL”	Required
SEQ_VAR1	Sequence Variable #1	Required
SEQ_VAR_TYPE2	Sequence Variable #2 Type “GLOBAL” or “LOCAL”	Required
SEQ_VAR2	Sequence Variable #2	Required
LABEL_NAME	LABEL to branch to upon condition being true.	Required

## **LABEL**

<b>Parameter Name</b>	<b>Description</b>	<b>Req./Opt.</b>
LABEL_NAME	Branch to Label Name	Required

## 2.6 NIPC and EC SASF Formats

SASF's produced by SCT and SOPC software are "mini-sASF's" and can have a limited SFDU and sequence headers. These files can then be run through seqgen, using the check\_sASF function, to produce a SASF with full SFDU and sequence headers. The full headers are required in order to place the file on the PDB. The mini-sASF file must contain the following:

```
DATA_SET_ID = SPACECRAFT_ACTIVITY_SEQUENCE_xxx
CCSD3RE00000ABXDEFGHNJPL3IF0M01300000001;
*SPAE_ID
$$EOH
$$EOD

(REQUESTS)

$$EOF
```

Where xxx = MAG, MOC, MOLA, TES, SCT

## 2.7 MGS - Special Payload Blocks

This section describes special payload blocks which are not described in the MGS block dictionary.

### PLOADM

The PLOADM block has a limit of 126 words, it is used for generating instrument memory loads (SCPDSC commands), where the command timing is not critical. The seqgen block, PLOADM, sets up a call to the seqtran macro MLOADP. The MLOADP macro contains OPEN\_TIME and CLOSE\_TIME parameters. This sets up a window seqtran uses to determine where to put the resultant SCPDSC commands in the SCMF. The basic algorithm is , seqtran takes all MLOADP calls in an SSF within an OPEN\_TIME and CLOSE\_TIME window and writes them as SCPDSC commands sequential to the SCMF starting at OPEN\_TIME. The spacing between the SCPDSC commands is based on the length of the SCPDSC load, the uplink data rate and message delay. All SCPDSC messages must be completed before CLOSE\_TIME. Other commands in the SSF take priority and may be interspersed with the SCPDSC commands.

## 2.8 MGS - Special Spacecraft Blocks

These blocks call special seqtran macros for managing multi-block data loads.

### **SAULOD** - Sun Centered Star Catalog Load Support Macro

Generates:

SAUCLB  
SAUCLC  
SAUCLE

### **SASLOD** - Standard Star Catalog Support Macro

Generates:

SASCLB  
SASCLC  
SASCLE

### **SAE1LD** - Secondary Emergency TLM Table 1 Load Support Macro

Generates:

SCAE1B  
SCAE1C  
SCAE1E

### **SAE2LD** - Secondary Emergency TLM Table 2 Load Support Macro

Generates:

SCAE2B  
SCAE2C  
SCAE2E

**SAENLD** - Secondary Emergency TLM Table Load Support Macro

Generates:

SCAENB  
SCAENC  
SCAENE

**SAMILD** - Secondary Mission TLM Table Load Support Macro

Generates:

SCAMIB  
SCAMIC  
SCAMIE

**SPE1LD** - Primary Emergency TLM Table 1 Load Support Macro

Generates:

SCPE1B  
SCPE1C  
SCPE1E

**SPE2LD** - Primary Emergency TLM Table 2 Load Support Macro

Generates:

SCPE2B  
SCPE2C  
SCPE2E

**SPENLD** - Primary Engineering TLM Load Support Macro

Generates:

SCPENB  
SCPENC  
SCPENE

**SPMILD** - Primary Mission TLM Table Load Support Macro

Generates:

SCPMIB  
SCPMIC  
SCPMIE

**SCSLOD** - Script Buffer Load Support Macro

Generates:

SCSBLB  
SCSBLC  
SCSBLE

**FSWLOD** - FSW Memory Load support Macro

Generates:

SCGNLD

### 3. Sequence Software Functions

#### 3.1 Core Software

##### 3.1.1 seqgen

Description:

 [User's Guide](#)

The **seqgen** command is the multi-mission sequence generation program.

Command Format:

**seqgen** <flags> <environment file>

Command Parameters:

usage: **seqgen** [[-abdefilmnoprstDNS] [<environment file>]] | [-b [-ano] <environment file>]

where:

Parameter Name	Description	Req./Opt.
flags	-a do not align commands to frame boundaries	Optional
	-b run in batch mode	Optional
	-d run in silence mode for reading and writing of files only	Optional
	-e display error report upon exit	Optional
	-f run in fast alone, modeler user interface is disable	Optional
	-i run in silence mode: no popup alerts or warnings	Optional
	-l auto-load files upon startup	Optional
	-m manually wait for modeling program to connect	Optional
	-n turn off modeling	Optional
	-o run expansion program alone, modeling is automatic off	Optional
	-p align from previous step, start plus delta(s) then align	Optional
	-r remove spillover activities in output conditions files	Optional
	-s execute one command at a time when running in Script Mode	Optional
	-t use standard time format in SSF (e.g. 1999-123T12:12:00.000)	Optional
	-D start in debug mode	Optional
	-N allow negative stimuli	Optional
	-S start in script mode	Optional
env file	The name of the environment file	Required

Example:

```
seqgen -dl p995a.env
```

### 3.1.2 seqtran - MGS Only

Don't try this at home.

Description:

 [Users Guide](#)

**SEQTRAN** consists of two distinct pieces used to support normal sequence translation. It provides the environment to perform MGS sequence translation, and the SEQTRAN Macro Library provides the mission specific macros required to actually accomplish sequence translation and memory management.

### 3.1.3 sts - (slinc)

Description:

 [SRD](#)

**Sequence Translation Subsystem (STS).** The STS is responsible for the translation of a Spacecraft Sequence in the form of a Spacecraft Sequence File (SSF) into a Command Packet File (CPF) for radiation to the spacecraft. In addition a binary UNIX file may be formatted into a CPF for transmission to the spacecraft. The primary program of the STS is the Spacecraft Language Interpreter and Collector (SLINC).

### 3.1.4 apgen

Description:

 [User's Guide](#)

**APGEN** is a planning tool component of the Mission Planning & Analysis Program Element of MGSO. The Mission Planning and Analysis Program Element is responsible for supplying the mission planning, mission sequencing, and spacecraft analysis portion of the MGSO ground data system. APGEN is used by mission planners and science planners to do resource-driven planning of mission activities.

### 3.1.5 seq\_review

Description:

 [User's Guide](#)

**seq\_review** is a MGSO sequence review tool. It allows the user to strip and format sequence products and to do rule checking. **seq\_review** requires a file descriptor in order to know how to interpret a particular file format. There is a file descriptor supplied by MGSO for the pef in \$SEQ\_EXE\_DIR/moso\_pef.fd.

Command Format:

**seq\_review <-config [file]> <-iconic> <-script [file]>**

Command Parameters:

Parameter Name	Description	Req./Opt.
-batch	run in batch mode (no windows)	Optional
-config [file]	use the specified file as a config file (instead of the default “seq_review.cfg” or whatever is pointed to by the \$seq_review_cfg environment variable)	Optional
-iconic	Will cause the windows to appear iconified. This allows users to run seq_review in batch mode (still requires a window manager to be running,however)	Optional
-script [file]	run script in specified file	Optional

Example:

**seq\_review**

## 3.2 Automation Tools

### 3.2.1 run.apgen

Description:

The command **run.apgen** is an automation script used to set up and run apgen for science plans. The first time run.apgen is envoked for a plan, it will generate a template APF and build apgen script files containing references to the adaptation, and any ancillary files required ( optg, DSN Allocation...), and pull the ancillary files from the FIS, if required. On subsequent runs using the APF, run.apgen will use the specified APF and script files generated from the previous run. The command line parameters (other than sequence\_name) listed as required are necessary only for the initial run.

Command Format:

```
run.apgen <-h> <-sc nnn> <-osc nnn> <-begin YYYY-DDDT00:00:00>
           <-end YYYY-DDDT00:00:00> <-reset reset_name>
           <-cond_seq cond_seq_file> <-apf apf_name> sequence_name
```

Command Parameters:

Parameter Name	Description	Req./Opt
-h	Print out Usage	Optional
-sc	Allows the user to specify the S/C Id Number, defaults to 84	Optional
-osc	Allows the user to specify the Orbiter S/C Id Number, defaults to 53	Optional
-begin	Specify the begin time of the sequence in MLT [dddThh:mm:ss] or UTC [yyyy-dddTThh:mm:ss]	Required
-end	Specify the end time of the sequence in MLT [dddThh:mm:ss] or UTC [yyyy-dddTThh:mm:ss]	Required
-reset	Specifies the name of the reset HSKP Sequence products (used by M01 Lander)	Optional
-cond_seq	Specifies the name of the sequence products to use for initial conditions.	Optioanl
-apf	Specify the name of an apf to read in	Optioanl
sequence_name	For a new file, This is the name of the output sequence products, For an existing file, it is the aaf name.	Required

Example:

Initial run of a plan

```
run.apgen -sc 116 -osc 120 -begin 8T00:00:00 -end 10T00:00:00 sol18_9
```

### 3.2.2 check\_sasf

Description:

The command **check\_sasf** is used to verify the format of a sASF. **check\_sasf** generates a SEQGEN environment file and then runs SEQGEN and FINDKEYS to check the SASF. **check\_sasf** returns a UNIX exit status of zero if the SASF contained no errors, if errors were found the UNIX exit status will be non-zero.

Command Format:

```
check_sasf -sc <spacecraft id number> -type <sequence type>
-cvf <cvf_file> <sasf name>
```

Command Parameters:

Parameter Name	Description	Req./Opt.
<b>-h</b>	Print out Usage	Optional
<b>-sc (nn)</b>	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
<b>-type</b>	The type of sequence:  Defaults to <b>nipc</b> <b>rtc, nipc, ec, ic or stored</b>	Optional
<b>-cvf</b>	Add reference to the named Context Variable File to the environment file.	Optional
<b>sasf name</b>	The name of the sASF file ( .sASF excluded).	Required

Example:

```
check_sasf -type nipc moc-c11-10
env_gen: Processing SASF file moc-c11-10.sasf
>> FINDKEYS Version 1.0
Did not find any keys.

moc-c11-10 passed SEQGEN check
```

### 3.2.3 wrap\_sasf

Description:

The command **wrap\_sasf** is used to wrap the sasf with the correct SFDU header.

Command Format:

**wrap\_sasf -sc <spacecraft id number> <sasf\_name>**

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Print out Usage	Optional
-sc (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
sasf name	The name of the sasf file (.sasf excluded).	Required

Example:

```
wrap_sasf -sc 94 moc_c11_10
```

### 3.2.4 send\_nipcec

Description:

The command send\_nipcec transfers the sASF file to the PDB and generates an EFRF and sends it to the nipcec daemon. The user should expect an E-mail receipt message from the nipcec daemon within 1 minute, and a completion message following that by 30 sec to 10 minutes, depending on nipcec system loading. If the sASF is a “mini-sASF” and does not have a SFDU header and Sequence header, then send\_nipcec will add the appropriate headers.

Command Format:

```
send_nipcec <-sc [nn]> <-bin [PDB_Bin]> <-u [uplink_rate]>
<-rad_open yyyy-dddThh:mm:ss>
<-rad_close yyyy-dddThh:mm:ss>
<-c [Comment]> <-testbed> sequence name
```

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Prints usage message	Optional
-bin [PDB_Bin]	The PDB Bin (MOC, MOLA, TES, MAG, SCT)	Required
-sc (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
-u [uplink_rate]	Uplink Bit Rate	Optional
-rad_open	Open of the uplink radiation window (yyyy-dddThh:mm:ss)	Optional
-rad_close	Close of the uplink radiation window (yyyy-dddThh:mm:ss)	Optional
-c [Comment]	Comment added to the EFRF	Optional
-testbed	Send the frf to the testbed sequence daemon	Optional
sASF_name	The name of the sASF file ( .sASF excluded).	Required

Example:

```
send_nipcec -bin SCT -sc 94 any_sASF
```

### 3.2.5 send\_ic

Description:

The command send\_ic transfers the sASF file to the PDB and generates an EFRF and sends it to the nipcec daemon. The user should expect an E-mail receipt message from the nipcec daemon within 1 minute, and a completion message following that by 30 sec to 10 minutes, depending on nipcec system loading. The send\_ic command notifies the nipcec daemon that the command is a S/C interactive command and whether to merge the command request with the background sequence.

Command Format:

```
send_ic <-sc nn> <-m> <-u [uplink_rate]> <-c [Comment]>
<-testbed> sequence name
```

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Prints usage message	Optional
sequence name	The name of the sASF file (.ASF excluded).	Required
-sc (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
-m	Merge with background sequence	Optional
-u [uplink_rate]	Uplink Bit Rate	Optional
-c [Comment]	Comment added to the EFRF	Optional
-testbed	Send the frf to the testbed sequence daemon	Optional
sASF_name	The name of the sASF file (.ASF excluded).	Required

Example:

<b>send_ic -sc 94 any_sASF</b>
--------------------------------

### 3.2.6 send\_frf

Description:

The command **send\_frf** generates an EFRF and sends it to the nipcec daemon. This command would be used if the sASF is already on the PDB. The user should expect an E-mail receipt message from the nipcec daemon within 1 minute, and a completion message following that by 30 sec to 10 minutes, depending on nipcec system loading.

Command Format:

```
send_frf <-sc nn> <-type [type] > <-C> <-m> <-u [uplink_rate]>
    <-c Comment> <-cvf cvf_name> <-aux_sasf sasf_name>
    <-optg optg_name> <-local> <-testbed> <-reset_seq_memory>
    <-rad_open yyyy-dddThh:mm:ss> <-rad_close yyyy-dddThh:mm:ss>
    <-target_dir> <-cfg> <-marci_delete><-sequence name
```

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Print out Usage	Optional
-sc (nn)	Spacecraft Id number ,i.e. MGS = 94, 95 Stardust = 29, 35, 157 M01 Orbiter = 53, 34	Required Genesis = 47, 54 Deep Space 1 = 30, 37 SIRTF = 79, 125
-type [type]	ec = Express Command nipc = Payload Command abm = Aerobrake Maneuver ifl = interactive file load	ic = Interactive Command drag = Aerobrake Drag Pass sci = Aerobrake Science nifl = non-interactive file load
-m	Manually merge ic with background sequence	Optional
-load_and_go	Specifies that the scmf should contain a command to activate the sequence as soon as it is loaded	Optional
-C	Contingency Command Flag	Optional
-c [Comment]	Comment added to the EFRF	Optional
-u [uplink_rate]	Uplink Bit Rate	Optional
-cvf [cvf_name]	Context Variable File to be used	Optional
-aux_sasf [sasf_name]	Add reference to the SASF to the EFRF.	Optional
-optg [optg_name]	OPTG File to be used for modeling	Optional
-local	Run the sequence manually, no frf is sent to sequence daemon	Optional
-testbed	Send the frf to the testbed sequence daemon	Optional
-reset_seq_memory (MGS Only)	Directs SEQTRAN to use the default SYMIN file for memory management purposes	Optional
-rad_open (nipc - only)	Open of the uplink radiation window (yyyy-dddThh:mm:ss) .	Optional
-rad_close (nipc - only)	Close of the uplink radiation window (yyyy-dddThh:mm:ss) .	Optional
-target_dir	Required for [-ifl], specifies the on-board directory, ie c:/cfg	Optional
-cfg	A list of cfgs to be appended to the uplink.	Optional
-marci_delete	For MARCI NIFL cmds, delete the file after transfer to the MARCI.	Optional
file_name	This is the sASF/file_load name to process	Optional

Example:

```
send_frf -sc 94 -type ic sasf_name
```

### 3.2.7 gen\_command

Description:

**gen\_command** is a high-level sequence system interface. It allows the user to start from scratch and generate all sequence output products, or to re-process an existing sequence request file (SASF). It can be run in “batch” mode on existing SASF’s, or seqgen can be invoked interactive to generate or update a sequence.

Command Format:

```
gen_command <-h> <-sc spacecraft_number> <-b batch> <-begin yyyy-dddTThh:mm:ss>
             <-end yyyy-dddTThh:mm:ss> <-sASF_bin sASF_bin> <-duration dddTThh:mm:ss>
             <-model> <-activate> <no_uplink> <-r uplink_bit_rate> <-oc 1|2|3>
             <-command command_name> <-parameters parameter_list>
             <-processor processor_field> <-satf satf_name> <-sASF sASF_name>
             <-bg_sASF bg_sASF_name> <-cvf cvf_name> <-smf smf_name> <-send_scmf>
             <-script> <-host> <-port> sequence_name
```

Command Parameters:

Parameter Name	Description	Req./Opt
-h	Print out Usage	Optional
-sc (nn)	Spacecraft Id number, i.e. MGS = 94, 95                    GENESIS = 47, 54 Stardust = 29, 35, 157        Deep Space 1 = 30, 37 M01 Orbiter = 53, 34          SIRTF = 79, 125	Required
-batch (-b)	When processing an existing sASF, do not bring up Seqgen editor	Optional
-begin	Begin time of sequence, defaults to NOW	Optional
-end	End time of sequence	Optional
-sASF_bin	SASF bin for the DATA_SET_ID. i.e. SCT,MOC.when generating a SASF from scratch	Optional
-duration	Duration of sequence, may be used instead of end time (defaults to 30 days)	Optional
-model	Turns modeling on in seqgen	Optional
-activate	Generate SEQ_BLOCK_START or ACTIVAT_SEQUENCE and merge into scmf	Optional
-no_uplink	Turns off the generation of an SSF, used for model only merge runs	Optional
-r	Uplink bit rate in bits/second	Optional
-oc	Output Packet Repeat Count, defaults to the value in slinc config file	Optional
-command	Command name	Optional
-parameters	Specify command parameters. This gets tricky with UNIX and double quotes or []. If a parameter of type string is input, then double quotes are required in the sASF. This can be accomplished by doing something like -parameter \"AEAPYZ\"'	Optional

More options Next page.....

Parameter Name (cont)	Description	Req./Opt
-processor	Processor field in seqgen	Optional
-satf	Specify the name of an satf to be used for uplink generation	Optional
-sASF	Name of an sASF to be merged for uplink generation	Optional
-bg_sASF	Name of a background sASF	Optional
-smf	Override default Spacecraft Model File	Optional
-send_scmf	Send the resultant scmf using scmf2sink or tci	Optional
-script	Use script file for seqgen	Optional
-host	Destination Host for scmf2sink	Optional
-port	Destination Port for scmf2sink	Optional
-server	Destination server for tci, defaults to uconx	Optional
-link	Destination link for tci, defaults to 3	Optional
seq_name	For a new file, This is the name of the output sequence products. For an existing file, it is the sASF name.	Required

Example:

```
gen_command -sc 94 stssrc
```

### 3.2.8 env\_gen

Description:

The command **env\_gen** is used to generate or update a SEQGEN environment file. If the **-u** option is used, **env\_gen** will update an existing environment file, otherwise it will scan an sasf and generate a new environment file, or for a new sequence for which no sasf exists. **env\_gen** determines the appropriate LIGHTTIME, OPTG, SCLK\_SCT, DSN\_VIEWPERIODS and DSN\_EIGHT\_WEEK\_SCHEDULE by querying the PDB to determine the newest file of the appropriate type which covers the time span of the sequence. If the file does not exist on the local SEQ workstation, the file is extracted from the PDB and placed in the appropriate directory. If there is no sasf file for the sequence **env\_gen** will prompt the user for the sequence start time, cut-off time and sequence title. The output environment file will be named sequence\_name.env.

Command Format:

```
env_gen <-u> <-sc sc_num> <-cvf cvf_file> <-type sequence_type> <- smf smf_name>
<-script script_name> <-optg optg_name> <-bg_sasf bg_sasf_name>
<-satf satf_name> <-sasf sasf_name> <-begin yyyy-dddThh:mm:ss>
<-end yyyy-dddThh:mm:ss> <-no_uplink> <sequence name>
```

Command Parameters:

Parameter Name	Description	Req./Opt.
<b>-h</b>	Print out Usage	Optional
<b>-type</b>	The type of sequence to process <b>nipc</b> - for NIPC command files <b>ec</b> - for EC files <b>ic</b> - for IC <b>stored</b> - for Stored Sequence files <b>rtc</b> - for real-time command files	Required
<b>-sc (nn)</b>	Spacecraft Id number, i.e. MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
<b>-u</b>	Update existing environment file	Optional
<b>-script</b>	The name of the seqgen script file	Optional
<b>-optg</b>	The name of the optg file	Optional
<b>-cvf</b>	The name of the context variable file	Optional
<b>-smf</b>	Override default Spacecraft Model File	Optional
<b>-bg_sasf</b>	The name of the background sasf	Optional
<b>-satf</b>	Specify the name of a satf file	Optional
<b>-sasf</b>	The name of an sasf to be merged for uplink generation	Optional
<b>-begin</b>	sequence begin time	Optional
<b>-end</b>	sequence cutoff time	Optional
<b>-no_uplink</b>	No uplink (ssf) is to be generated	Optional
<b>sequence name</b>	The name of the sasf file (.sasf excluded).	Required

Example:

```
env_gen -sc 94 -type stored p995a
```

### 3.2.9 seq\_translate

Description:

The command **seq\_translate** is used to turn a ssf into a scmf. The script will determine the S/C number from the ssf header.

Command Format:

```
seq_translate <-r bit rate> <-d message delay> <-oc> <-activate> <-no_GCMD>
<-p prev_seq> <-ssf_list ssf_name> <-cfg cfg_name> <sequence id>
```

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Print out Usage	Optional
-d (message delay) [MGS only]	The delay in seconds (Hex) between scmf messages Default =1 second	Optional
-r	Uplink bit rate, i.e. 2000 bps                    1000 bps 500 bps                    250 bps 125 bps (Default)        62.5 bps 31.25 bps                15.625 bps 7.8125 bps	Optional
-oc	Output Packet Repeat Count, defaults to the value in slinc config	Optional
-activate	Generate SEQ_BLOCK_START or ACTIVAT_SEQUENCE and merge into scmf	Optional
-no_GCMD	Don't generate a gcmd	Optional
-p (prev_seq)	Previous sequence id, for MGS memory management.	Optional
-ssf_list	A ssf to be appended to the uplink	Optional
-cfg	A cfg to be appended to the uplink	Optional
sequence id	The name of the ssf file ( .ssf excluded).	Required

Example:

```
seq_translate p995a
SEQTRAN processing ssf: p995a
SEQTRAN completed successfully
```

### 3.2.10 file\_load

Description:

The command **file\_load** is a utility to prep and generate uplink products for flight software loads.

Command Format:

```
file_load <-h> <-r bit rate> <-sc spacecraft_id> <-p path> <-sr> <-load_to_tmp>
<-file_type type> <-split_uplink> <-seq_id MSP_seqid> <-seqnam DS1_seqname>
<-seqnum DS1_seqnum> <-cmd_offset hh:mm:ss> <-activate> <-marci> <-pmirr>
<-marci_delete> <file_name>
```

Command Parameters:

Parameter Name	Description	Req./Opt
-h	Print out Usage	Optional
-sc (nn)	Spacecraft Id number ,i.e. Stardust = 29, 35, 157   Deep Space 1 = 30, 37 Genesis = 47, 54   M01 Orbiter = 53, 34   SIRTF = 79,125	Required
-r	Uplink bit rate, i.e. 2000 bps      1000 bps      500 bps 250 bps      125 bps (Default)      62.5 bps 31.25 bps      15.625 bps      7.8125 bps	Optional
-p	Directory path/file_name to load the file on-board the spacecraft	Required
-sr	A different name for the scmf, defaults to file_name	Optional
-load_to_tmp	For small files, loads file to /tmp and then copies to final directory	Optional
-file_type	For MPF type file loads NORMAL SCRIPT PARAM, defaults to NORMAL	Optional
-split_uplink	For large files which are split into multiple uplinks, this will create a separate scmf/gcmd for the reconstruction sequence	Optional
-seq_id	If split_uplink is used, this parameter specifies the name of the MSP sequence which will put the file back together	Optional
-seqnam	If split_uplink is used, this parameter specifies the name of the DS1 sequence which will put the file back together defaults to UTIL	Optional
-seqnum	If split_uplink is used, this parameter specifies the number of the DS1 sequence which will put the file back together defaults to 4095	Optional
-cmd_offset	Specifies the timing offset between commands which put a split file back together, defaults to 00:00:02	Optional
-activate	If the split_uplink is used, this adds an ACTIVAT_SEQUENCE command to the sequence scmf	Optional
-marci	Perform a MARCI_FILE_LOAD	
-pmirr	Perform a PMIRR_FILE_LOAD	
-marci_delete	Delete file from marci directory after MARCI_FILE_LOAD	
file_name	This is the UNIX file name to process into an uplink product	Required

Example:

```
file_load -sc 29 -p /tables priority_table
```

### 3.2.11 printscmf

Description:

The **printscmf** command dumps a SCMF file in ASCII to the screen.

Command Format:

**printscmf <scmf file>**

Command Parameters:

Parameter Name	Description	Req./Opt.
scmf file	The name of the scmf file.	Required

Example:

```
printscmf nc0001.scmf
>> PRINTSCMF Version 1.0

scmf nc0001.scmf is SFDU wrapped

Header information:
    Filename      : NC0001
    Preparer      : Steven Wissler
    File Size     : 634
    File Header Size : 384
    Mission ID   : 5
    Spacecraft ID: 94
    Reference Number : 0
    Bit 1 Rad. Time :
    Bit Rate       : 5: 125 bits/second
    Comment        :
    Creation Time : 95-254/13:44:42
    Title          : MO
    Seqtran Version: SEQTRAN -MGS V20MGS.0 Jan 23 07:09 PST 1995
    Macro Version  : MARS GLOBAL SURVEYOR           JULY 26, 1995
    File Checksum  : 22928
```

Messages:

```
Message : 001 -- Length : 00816
Transmission Start : 93-179/16:03:37.900
Open Window       :
Close Window      :
Message Comment   :
Message Checksum  : 63797
Message Bits      :
    aaaa  aaaa  aaaa  aaaa  aaaa  aaaa  aaaa  aaaa  aaaa  aaaa
    aaaa  f238  ca00  010c  0a68  0001  001a  d498  c006  6101
    0000  a0bc  425b  1961  0021  240a  1a00  1e00  213c  4f4d
    2043  3143  0c20  316f  6c64  619e  3120  2030  7865  9063
    6574  756e  69e6  2e67  c40a  00eb  d6ff  ff00  0058  2406
    aaaa
End of file.
```

### 3.2.12 revinit\_gen

Description:

**revinit\_gen** generates a revtran initialization file for the input sequence. The output file is named sequence\_name.revinit

Command Format:

**revinit\_gen <sequence name>**

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the sequence to process	Required

Example:

```
revinit_gen s01b2
```

### 3.2.13 run.revtran

Description:

**run.revtran** is an automated script for running revtran. It generates a revtran initialization file and then runs revtran with the -batch and -native options. The output file is named sequence name.rpef

Command Format:

**run.revtran <sequence name> [print]**

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the sequence to process	Required
print	If specified produces the -print option on revtran	Optional
native	Output the datawords in the SEQGEN native types. If not specified the data words will be in HEX	Optional

Example:

```
run.revtran s01b2
```

### 3.2.14 revtran

Description:

**revtran** is a SCMF reverse translater. It generates a “PEF-like” file (RPEF) from a SCMF file. **revtran** requires a “set-up” file to specify all of the input and output files. The program **revinit\_gen** will generate the required file for an input sequence. See the revtran users guide for a full description.

Command Format:

**revtran** <-init init\_file> <-batch> <-trans> <-print> <-native>

Command Parameters:

Parameter Name	Description	Req./Opt.
-init init_file	The name of the file containing a list of revtran input/output files.	Required
-batch	Run revtran in batch mode	Optional
-trans	Translate the SCMF	Optional
-print	Print the SCMF to the log file	Optional
-native	Output the datawords in the SEQGEN native types. If not specified the data words will be in HEX	Optional

Init File Example:

```

#
# File:          C1A.revinit
# Description:  REVTRAN Initialization File (RIF)
#               All values must be specified starting in column 38.
# Version:      @(#)RELEASE 2.0a2
# Modified:     20 Aug 1993
#
A) SCMF (Input)           : s01a2.scmf
A) SCMF (Input)           : s01b2.scmf
A) SCMF (Input)           : s01c2.scmf
A) SCMF (Input)           : s01d2.scmf
A) SCMF (Input)           : s01e2.scmf
A) SCMF (Input)           : s01f2.scmf
B) RPEF (Output)          : s01-2.rpef
C) Spacecraft Memory [Uninit/Init] : Init
D) MIF (Input)            : ../launch/launch.separation.test.mif
E) MIF (Output)           : s01-2.mif
F) OWLT (Input)           : /seq/lrf/lrm_c_961105-970916_geo_test
G) SCLK Initial Value    : 0
H) SCLK_SCET (Input)      : /seq/sclk/SCLK_SCET.019
I) Mnemonics (Input)      : /seq/bin/mnemonics.L1.0.h
J) Run Log (Output)        : s01-2.revtran.log

```

Example:

```
revtran -init s01.revinit -batch -trans -native

>> MOSORT Version 8.2

Input file  -> /usr/tmp/baaaa08348
Output file -> s01-2.rpef

Starting sort on Fri Dec 29 09:08:58 1995

Phase 1: Dividing file...
Phase 2: Time-stamping events...
Phase 3: Sorting events...
Phase 4: Removing time-stamps...
Phase 5: Rewriting SEQTRAN SETTER's...
Phase 6: Updating filename in header...
Phase 7: Integrating file...

Finishing sort on Fri Dec 29 09:09:02 1995

REVTRAN 1.0

Jet Propulsion Laboratory
California Institute of Technology

Copyright (C) 1993, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.
```

### 3.2.15 revcompare

Description:

**revcompare** is a set of scripts and programs which compare a pef and rpef.

Command Format:

**revcompare** [options] <rpef> <pef>

Command Parameters:

Parameter Name	Description	Req./Opt.
options	-p : Ignore processor field	Optional
rpef	Name of RPEF file	Required
pef	Name of PEF file	Required

Example:

```
revcompare -p launch.triggers.rpef launch.test.pef

Note: Lines from the RPEF are marked with a '>'.
      Lines from the PEF are marked with a '<'.

< 0531633616:000 1996-310T03:59:59.846 DCMD,mpphase_d,NONE,LAUNCH,1996-310T04:00:00.000;
>
< 0531636376:000 1996-310T04:45:59.846 DCMD,mpphase_d,NONE,INNCRU,1996-310T04:46:00.000;
>
< 0531849556:000 1996-312T15:58:59.834 DCMD,SAGANS,RTC;
>
```

### 3.2.16 run.revcompare

Description:

**run.revcompare** is a script which runs **revcompare** with a standard set of options on an input sequence.

Command Format:

**run.revcompare <sequence name>**

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the sequence to process	Required

Example:

```
run.revcompare s01-2
```

### 3.2.17 get\_pef\_errors

Description:

**get\_pef\_errors** is a script which runs **seq\_review** to strip and format the error and warning messages in the pef. The output file containing the pef errors/warnings is named sequence\_name.pef\_errors.

Command Format:

**get\_pef\_errors <sequence name>**

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the sequence to process	Required

Example:

```
get_pef_errors p995a
```

### 3.2.18 pef\_display

Description:

**pef\_display** is a script which runs seq\_review to strip and format useful information from the pef. There are two time format options, either an epoch relative time or SCET can be displayed in the second time column.

Command Format:

```
pef_display <-e epoch> <-l light_time_file> <-b> <-soe> <-block> <-version> soe_ver>
<-No_DSN> <-No_Errors> <-Science> <-SciTimes> <-Power> <-LIDAR>
<-Lander> <-Bob> <-MIPL> <-EDL> <-EDL_Interface> <-TouchDown>
<-UHF_Passes> <-SSI> <-CMD_Only> <-Compare> <-DKF> <-Plot> <pef name>
```

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Prints usage message	Optional
-b	Runs pef_display in batch mode and creates the file pef_name.pef.display	Optional
-l <ltf filename>	Allows the user to override the light-time file specified in the pef header	Optional
-e	Allows the user to specify an epoch and produce an alternate time column with epoch-relative duration's. Default for the alternate time column is S/C Event Time	Optional
-e START	sets the epoch to the time of the first command in the sequence.	Optional
-e RELATIVE	sets the alternate time format to mimic a time-relative block, this is convenient for relative time sequence blocks	Optional
-soe	Generates a soe -like display	Optional
-SFOS	Generates a sfos	Optional
-block	Generates a relative time block display	Optional
-version	Version number for dkf and rsoe, defaults to 00	Optional
-No_DSN	Generates a strip with no DSN activities	Optional
-No_Errors	Generates a strip without errors and warnings	Optional
-Science	Generates a Science strip.	Optional
-SciTimes	Generates a List of M98 Lander science sequence start times.	Optional
-Power	Generates a M98 Lander power model strip.	Optional
-LIDAR	Generates a LIDAR strip.	Optional
-Lander	Generates a Lander On/Off strip.	Optional
-Bob	Generates a Bob Denise Format.	Optional
-MIPL	Generates a MIPL plot.	Optional
-EDL	Generates a EDL strip.	Optional
-EDL_Interface	EDL Interface Time.	Optional

More options Next page.....

Parameter Name	Description	Req./Opt.
-TouchDown	TouchDown Time.	Optional
-UHF_Passes	Generates a Lander UHF Passes strip for MCO MAPGen.	Optional
-SSI	Generates a SSI strip.	Optional
-CMD_Only	Generates a CMD only strip.	Optional
-Compare	Generates a format suitable for diffing 2 pefs	Optional
-DKF	Generate a DSN Keyword File	Optional
-Plot	Generate a plot	Optional
pef_name	This is the name of the pef file to be displayed.	Required

Example:

```
pef_display -b -soe -version 01 p995a
```

Note: See Appendix A - for pef\_display screen dumps, using various options.

### 3.2.19 **pdb\_store**

Description:

**pdb\_store** adds all appropriate stored sequence products to the PDB

Command Format:

**pdb\_store** <sequence name>

Command Parameters:

Parameter Name	Description	Req./Opt.
Sequence name	The name of the sequence to process	Required

Example:

**pdb\_store p995a**

### 3.2.20 notify\_act

Description:

**notify\_act** fills out the ECRF form on the Automated Command Tracker (ACT) with all appropriate file names added to the PDB

Command Format:

```
notify_act <-h> <-b> <-file_name [input sasf name]>  
<-form_name [form name]> seq_name
```

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Prints usage message	Optional
-b	Do not bring up the ACT GUI	Optional
-file_name input sasf name	For NIPC,IC and EC, this is the input SASF name	Optional
-form_name form name	This is the name of an ECRF form to invoke, if not supplied, the default for the S/C will be used. This field would only be used when updating an existing ECRF	Optional
sequence name	The name of the sequence to process	Required

Example:

```
notify_act -b p995a
```

### 3.2.21 gen\_dsn\_config\_codes

Description:

**gen\_dsn\_config\_codes** is a perl script that converts the dsn configuration codes (provided by the DSN) to the format that SEGS understands. This script only needs to be run if a new dsn configuration file (DSN\_CONFIGURATIONS\_CODES) is delivered, due to new codes in the DSN\_EIGHT\_WEEK\_SCHEDULE. If changes to the new dsn configuration file are for new equipment added to the station, **gen\_dsn\_config\_codes** may need to be updated.

Command Format:

**gen\_dsn\_config\_codes <input file > dsn\_cfg\_codes**

Command Parameters:

Parameter Name	Description	Req./Opt.
Input file	The name of the downloaded DSN_CONFIGURATIONS_CODES file	Required

Example:

**gen\_dsn\_config\_codes <input file > dsn\_cfg\_codes**

Note: The output file must always be dsn\_cfg\_codes.

### 3.2.22 run.dsn.config

Description:

**run\_dsn\_config** is a perl script that converts the SEGS formatted, dsn configuration file (dsn\_cfg\_codes) that is generated by running perl script **gen\_dsn\_config** to a format that seqgen understands.

Command Format:

**run\_dsn\_config <spacecraft\_number> dsn\_cfg\_codes**

Command Parameters:

Parameter Name	Description	Req./Opt.
Spacecraft (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
Input file	The name of the SEGS formatted DSN configuration file	Required

Example:

**run\_dsn\_config 94 dsn\_cfg\_codes**

Note: The output file will always be new.dsn\_config.csv

### 3.2.23 get\_seqtran\_info

Description:

The **get\_seqtran\_info** command strips the seqtran runlog for ERRORS and WARNINGS. Also included in this strip is the SCMF size, words used ,words remaining, starting memory location (in HEX) and header information.

Note: If generating a multi-part sequence, **get\_seqtran\_info** must be generated for each ssf generated.

Command Format:

**get\_seqtran\_info <sequence name >**

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the ssf file (.ssf excluded).	Required

Example:

```
get_seqtran_info p995a
```

### 3.2.24 run\_soe

Description:

The **run\_soe** command starts scripts **gen\_soe\_dkf** and **finish\_soe\_dkf**, these scripts generate the Sequence of Events files: tsoe - (text formatted - soe), rsoe.ps - (postscript formatted - soe) and the DSN Keyword File DKF. After the files are generated they are placed on the PDB and the MSOP Planning and Sequencing Homepage. The DKF is sent to the DSN file server.

Command Format:

**run\_soe <sequence name> < soe\_version\_number> <spacecraft\_number>**

Command Parameters:

Parameter Name	Description	Req./Opt.
-h	Prints usage message	Optional
-sc (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required
-ver (nn)	Specifies the SOE version number	Required
-fdsn	Run, ftp_dsn (this will send the DKF to the DSN server)	Optional
-fdsn_file filename	Allows you to attach special instructions to the email when the DKF is placed on the DSN server. Don't forget to build your text file first, before starting <b>run_soe</b> .	Optional
sequence name	The name of the pef file to process (.pef excluded).	Required

Example:

```
run_soe -sc 94 -ver 03 mm003c
```

mini soe\_procedures  
 [run\\_soe\\_details](#)  
 [run\\_soe](#)

### 3.2.25 gen\_soe\_dkf

Description:

**gen\_soe\_dkf** runs **pef\_display** and **correct\_transmitter\_min\_dur**. **pef\_display** generates the Sequence Of Events products: tsoe - (text formatted SOE), rsoe.ps (postscript formatted SOE ) and DSN Keyword File (DKF). **correct\_transmitter\_min\_dur** scans the tsoe for “TXFR on less than “ warnings.

Command Format:

**gen\_soe\_dkf** <sequence name> <soe\_version\_number>

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the pef file (.pef excluded).	Required
soe_ver	soe version number [nn]	Required

Example:

**gen\_soe\_dkf p995a 01**

### 3.2.26 correct\_transmitter\_min\_dur

Description:

**correct\_transmitter\_min\_dur** scans tsoe for “TXFR on less than” warnings, if any are found, runs seqgen including a sasf that delays the Transmitter ON. **pef\_display** is then re-run twice once, to generate a SOEs: tsoe - (text formatted - soe) then to generate the DKF.

Command Format:

**correct\_transmitter\_min\_dur** <sequence name> <soe\_version\_number>

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the pef file ( .pef excluded).	Required
soe_ver	soe version number [nn]	Required

Example:

```
correct_transmitter_min_dur p995a 01
```

### 3.2.27 **finish\_soe\_dkf**

Description:

**finish\_soe\_dkf** completes the SOE process, by placing SOE & DKF products on the PDB, converts postscript formatted SOE to PDF format, sends – PDF formatted SOE to the PST Homepage, runs **notify\_soe**, and runs - **ftp\_dsn**.

Note: If **finish\_soe\_dkf** is started by a non-nipcec account, user will be prompted to run **notify\_soe** and **ftp\_dsn**.

Command Format:

**finish\_soe\_dkf** <sequence name> <soe\_version\_number> <spacecraft\_number>

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the pef file (.pef excluded).	Required
soe_ver	soe version number [nn]	Required
-sc (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required

Example:

```
finish_soe_dkf p995a 01 94
```

### 3.2.28 notify\_soe

Description:

**notify\_soe** is run after the SOE is placed on the Homepage, it sends email to people listed on the notify\_soe\_mail\_list. Included in this email is the SOE filename, SOE file header and location of homepage. The message\_filename option is used so special instructions can be added to the email.

Command Format:

**notify\_soe <sequence name><soe\_version\_number><message\_filename>**

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the pef file ( .pef excluded).	Required
soe_ver	soe version number [nn]	Required
message_filename	Flme name containing special instructions	Optional

Example:

```
notify_soe p995a 01
```

### 3.2.29 ftp\_dsn

Description:

**ftp\_dsn** - sends the DSN keyword file (DKF) to the DSN server and sends email to people listed on the ftp\_dsn mail\_list. Included in this email is the DKF filename, DKF file header and location of homepage. The message\_filename option is used so special instructions can be added to the email.

Command Format:

**ftp\_dsn** <sequence name> <soe\_version\_number> <message\_filename>

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the pef file ( .pef excluded).	Required
soe_ver	soe version number [nn]	Required
message_filename	Flme name containing special instructions	Optional

Example:

```
ftp_dsn p995a 01
```

### 3.2.30 ftp\_2homepage

Description:

**ftp\_2homepage** - sends the rsqe.pdf and/or the DKF to the PST homepage. User will be prompted as to file type to send to homepage

Command Format:

**ftp\_2homepage** <sequence name> <soe\_version\_number> <spacecraft\_number>

Command Parameters:

Parameter Name	Description	Req./Opt.
sequence name	The name of the pef file (.pef excluded).	Required
soe_ver	soe version number [nn]	Required
-sc (nn)	Spacecraft Id number ,i.e.  MGS = 94, 95      Genesis = 47, 54 Stardust = 29, 35, 157      Deep Space 1 = 30, 37 M01 Orbiter = 53, 34      SIRTF = 79, 125	Required

Example:

```
ftp_2homepage p995a 01 94
```

## **4. Sequence Daemon Functions**

The Sequence Daemon functions are used to monitor and control the automated sequence processing.

### **4.1 Sequence Daemon**

The Sequence Daemon is a UNIX Cron daemon which is used to periodically wake up and read selected Electronic File Release Form e-mail from authorized users. The EFRF is processed for command requests. Any valid command requests are submitted to the UNIX batch queue for processing. The user is notified upon receipt and completion. Successful command requests are then forwarded to the ACT for disposition. (See Figure 4.1)

### **4.2 auto\_seq\_status**

This is a GUI interface to the Sequence Daemon functions. It allows the user to monitor the status of command requests and to control the operation and configuration of the Sequence Daemon. (See Figure 4.2).

#### **4.2.1 Display Screen**

The screen is split into 2 sections. The top section allows the user to turn the daemon on/off and to set the wake-up interval and the display refresh rate. The bottom section displays the status of command requests. The top panel shows jobs which have been completed, the middle panel shows the status of jobs currently processing and the bottom panel shows the status of jobs in the queue waiting to be processed. The user can examine any file generated by the sequence process by selecting a request in either of the top two panels by clicking on the appropriate status line with the mouse cursor. This will bring up the emacs editor with a directory listing of the files for that request. The user can then click on the file to be examined with mouse button 2 and emacs will open up that file for examination.

#### **4.2.2 Menu Items**

##### **4.2.2.1 File**

###### **4.2.2.1.1 Exit**

This button terminates the monitor and control program.

#### **4.2.2.2 Options**

##### **4.2.2.2.1 Sound**

This button will bring up the Openwindows gaintool application to allow the user to control the volume level of file arrival notifications.

##### **4.2.2.2.2 Edit User Table**

This button will bring up the emacs editor to allow the user to edit the user configuration table. This table contains the user accounts/nodes/bins which the daemon uses to determine who is allowed to send command requests.

##### **4.2.2.2.3 Set Default Uplink Rate**

This button allows the user to set the default uplink bit rate. This can be overridden in the command request( See send\_frf and send\_ic).

### 4.3 start\_auto\_sequence

Description:

This is the command line interface to start the sequence daemon. The user may specify the wake-up rate in minutes. It defaults to 1 minute. This function can be accessed via the monitor and control GUI. start\_auto\_sequence will not start the Daemon if the user is not authorized to run the Daemon. start\_auto\_sequence will syncronise the IC,EC and NIPC sequence ID's with the PDB. It will chose a sequence ID which is the larger of:

- one greater than the largest sequence ID on the PDB.
- The sequence ID in the seq\_id config file.

Command Format:

**start\_auto\_sequence <wake\_up\_rate>**

Command Parameters:

Parameter Name	Description	Req./Opt.
wake_up_rate	The interval in minutes for the daemon to wake up and process requests. Defaults to 1 minute.	Optional

Example:

```
start_auto_sequence 5
```

```
TMOD Automated Sequence Processing started on: mgseq5 at Wed Mar 17 06:58:45 UTC 1999.
```

#### 4.4 stop\_auto\_sequence

Description:

This is command line interface to stop the sequence daemon. start\_auto\_sequence will not start the Daemon if the user is not authorized to run the Daemon. This function can be accessed via the monitor and control GUI.

Command Format:

**stop\_auto\_sequence**

Command Parameters:

None:

Example:

**stop\_auto\_sequence**

TMOD automated sequence processing stopped on: mgseq5 at Wed Mar 17 05:00:45 UTC 1999.

**Figure 4.1**  
**Automated Sequence Processor - Block Diagram**

## Automated Sequence Process Data Flow

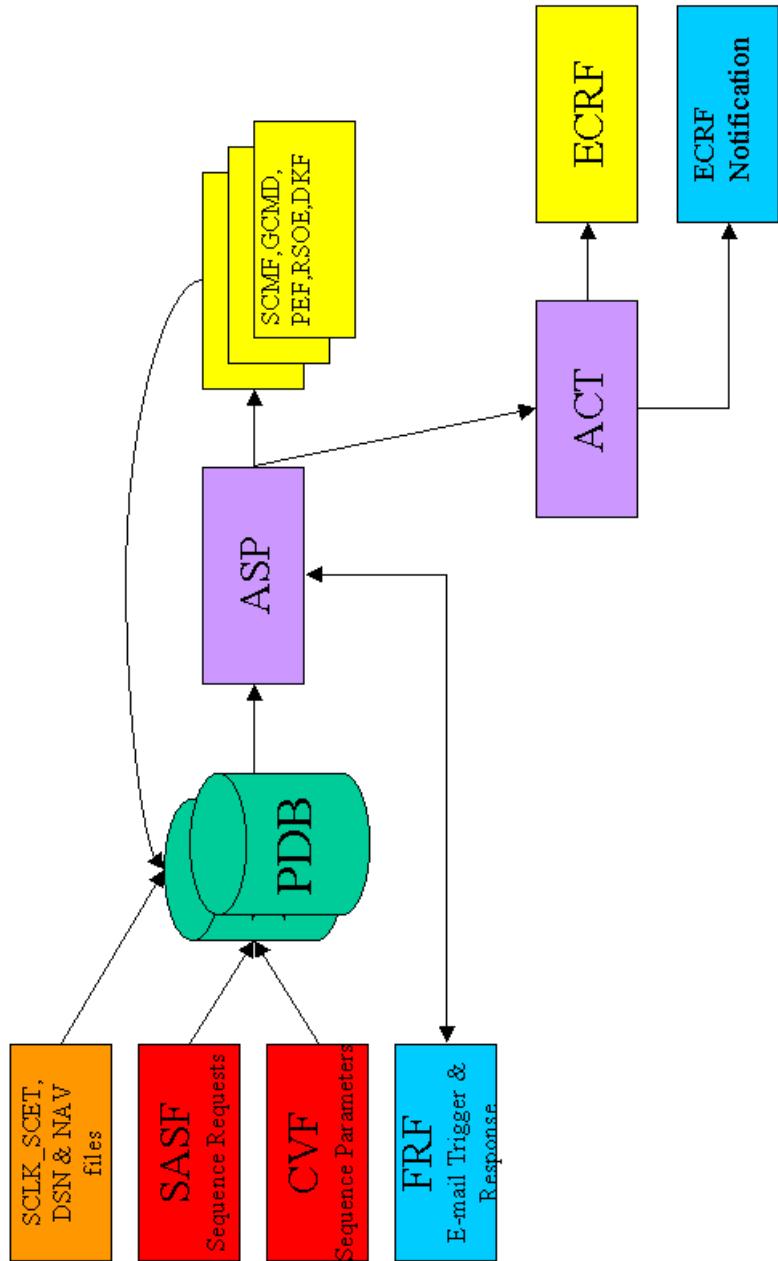
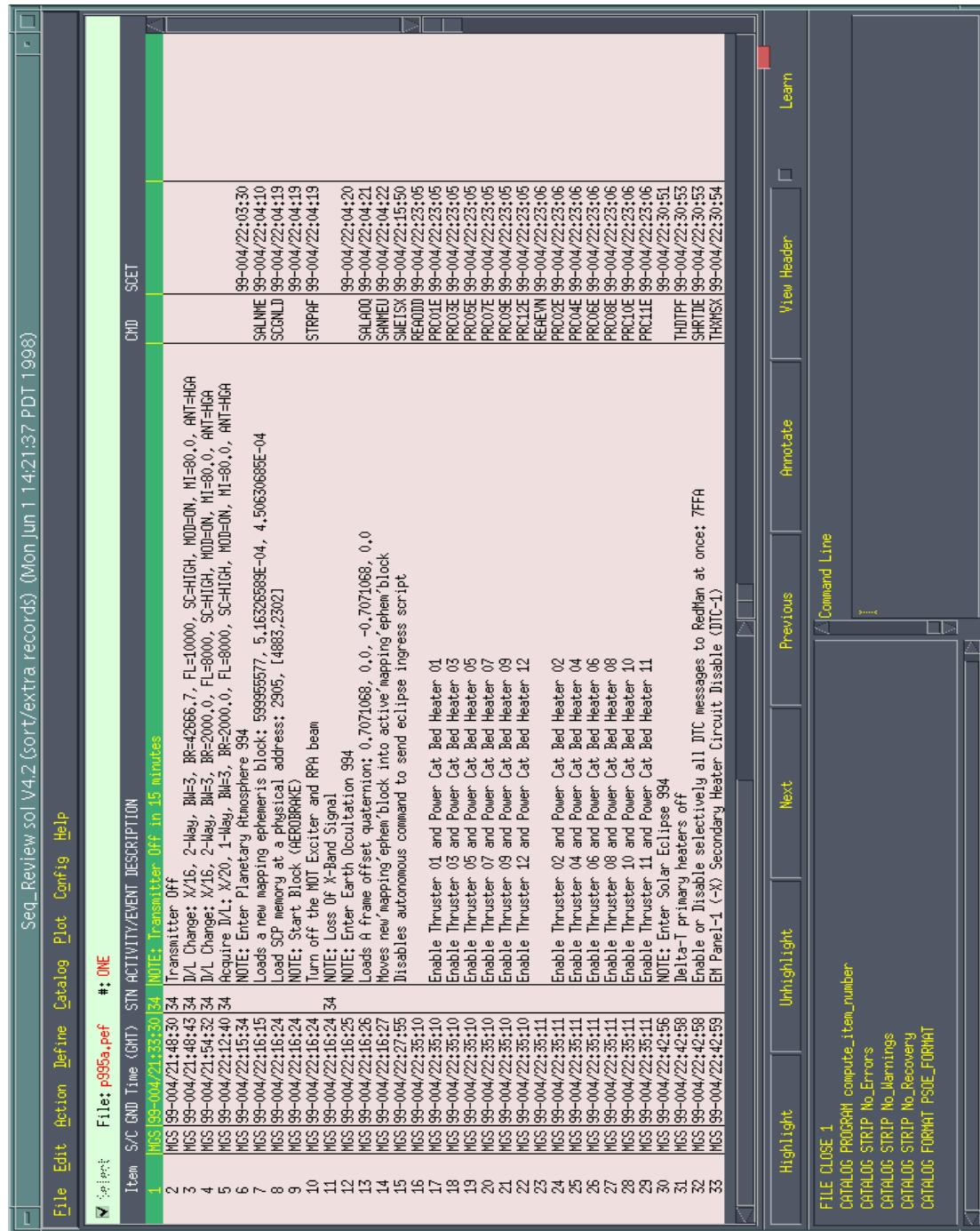


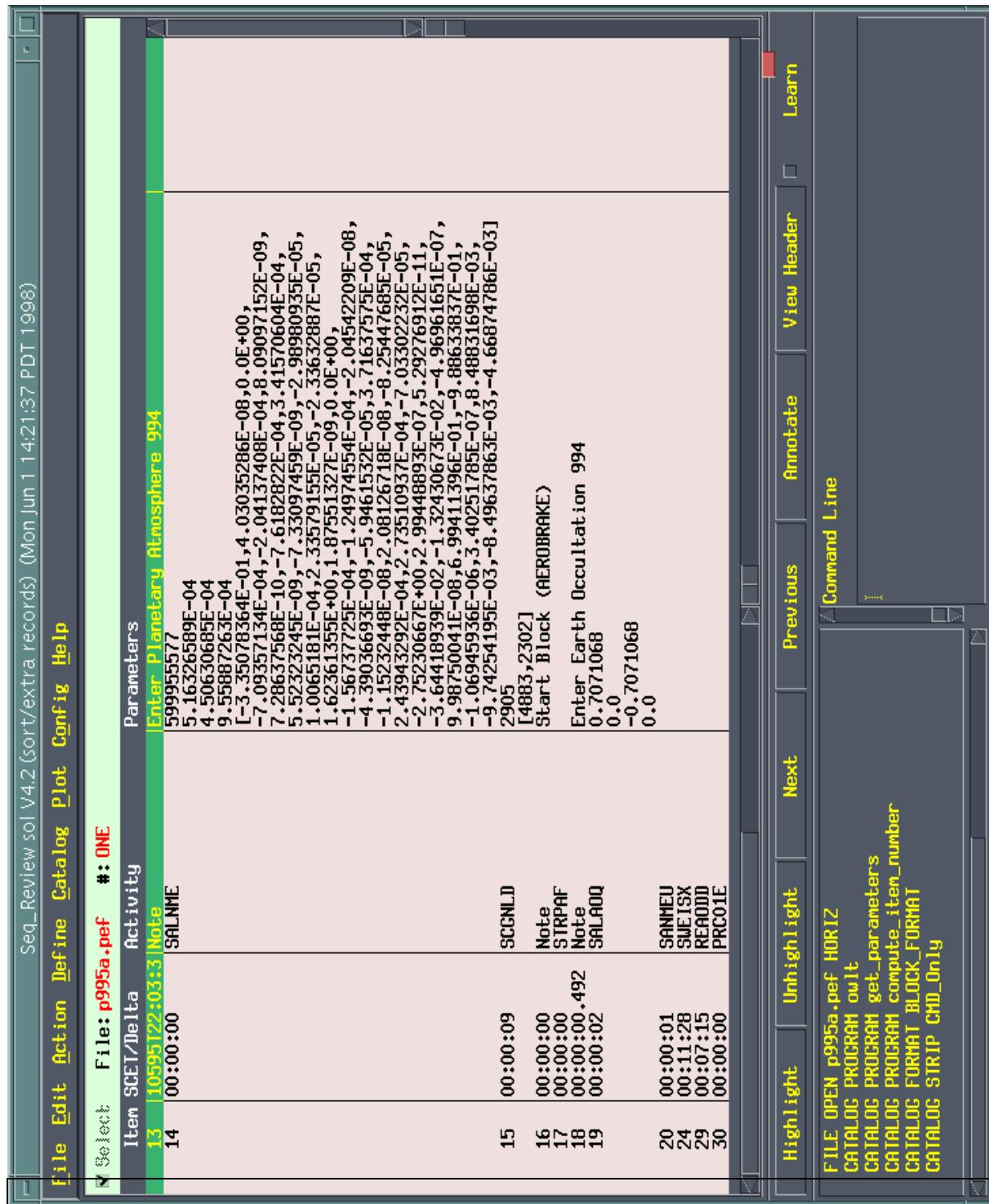


Figure 4.2  
Automated Sequence Monitor & Control GUI

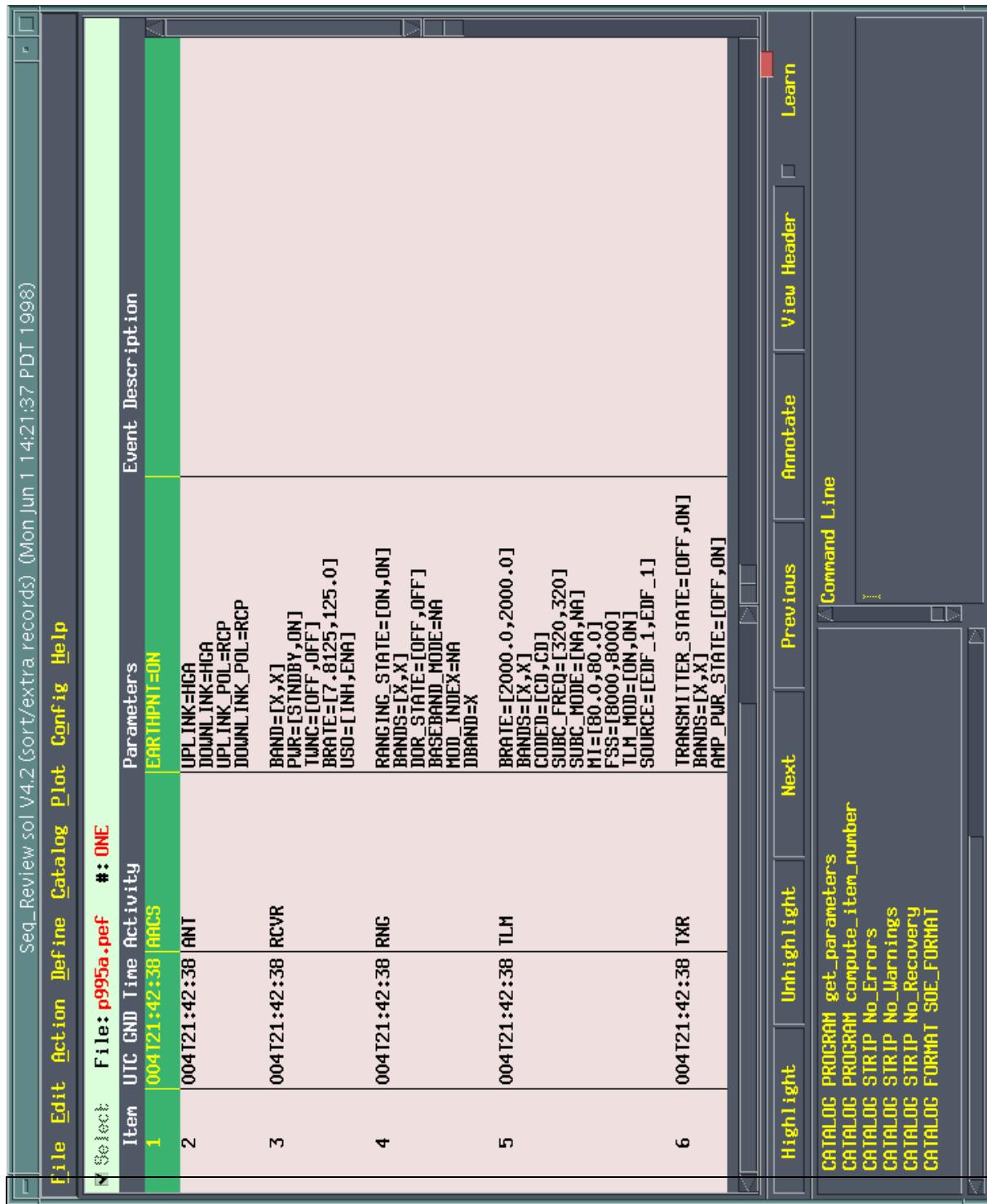
## 5. Appendix A



**Figure A.1**  
**Option Used [ -soe ]**  
**Example: pef\_display -soe p995a**



**Figure A.2**  
**Option Used [ -block ]**  
Example: **pef\_display -block p995a**



**Figure A.3**  
**Option Used [ -Science ]**  
**Example: pef\_display -Science p995a**

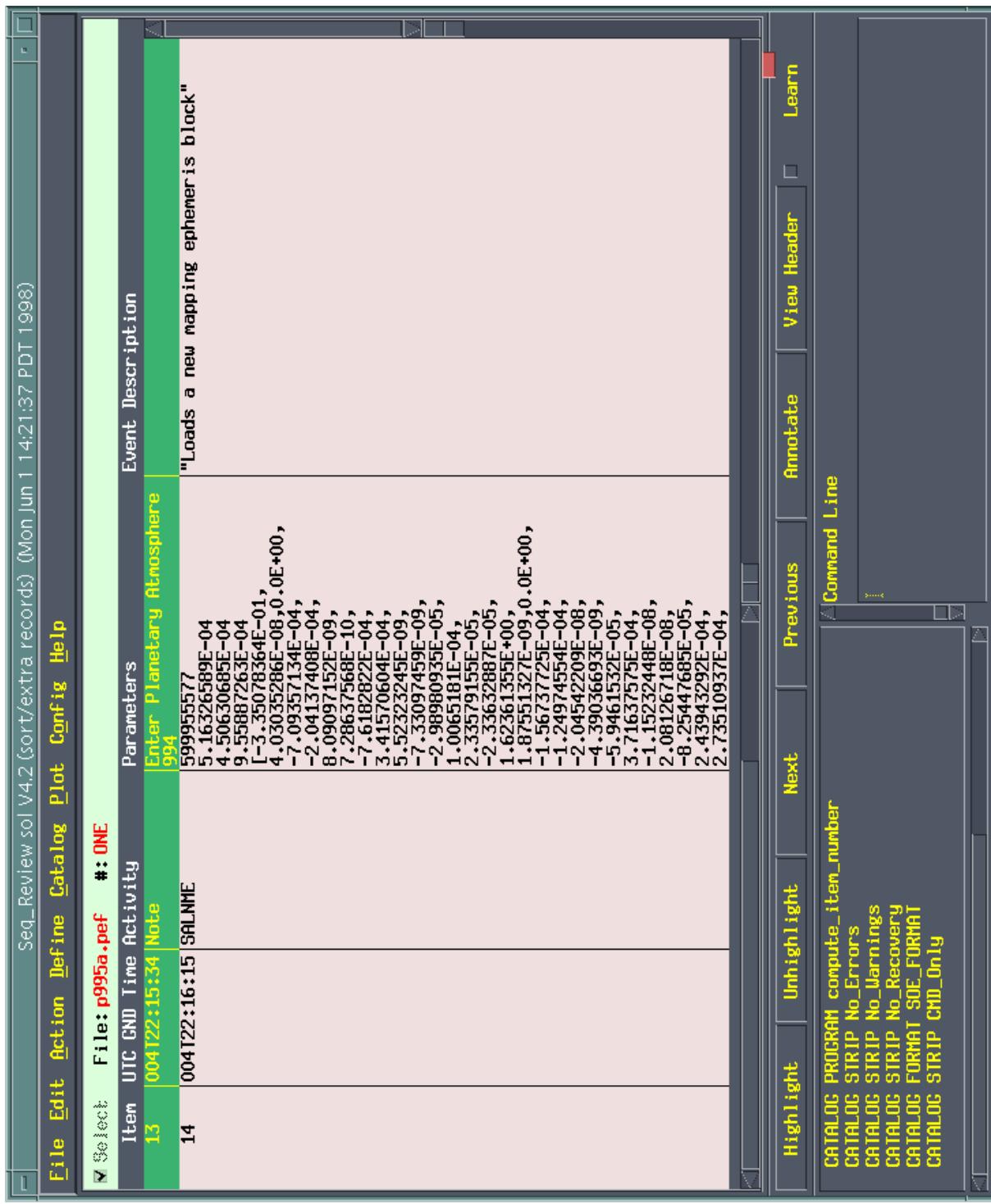
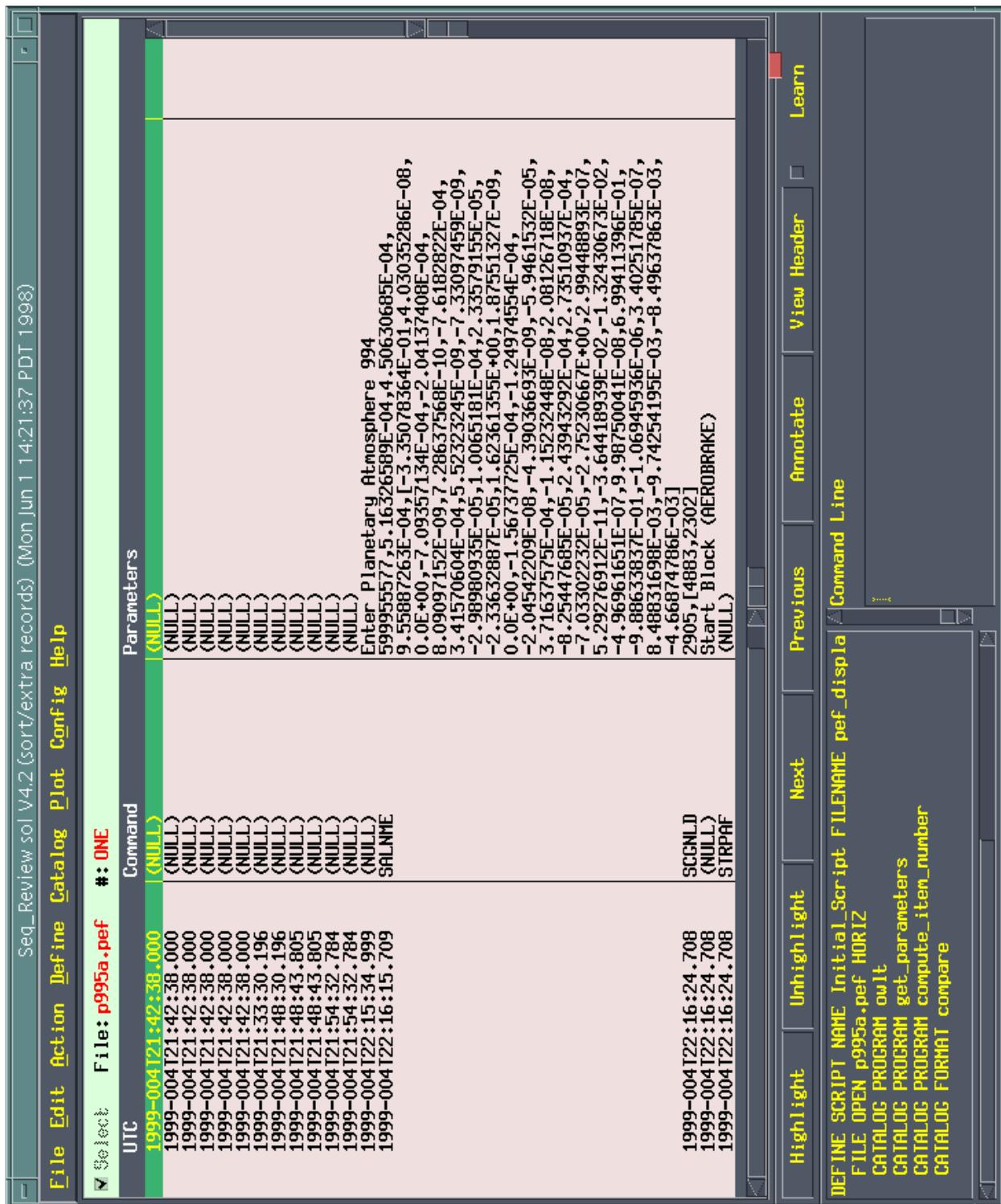


Figure A.4  
Option Used [ -CMD\_Only ]  
Example: **pef\_display -CMD\_Only p995a**



**Figure A.5**  
**Option Used [-Compare]**  
Example: **pef\_display -Compare p995a**

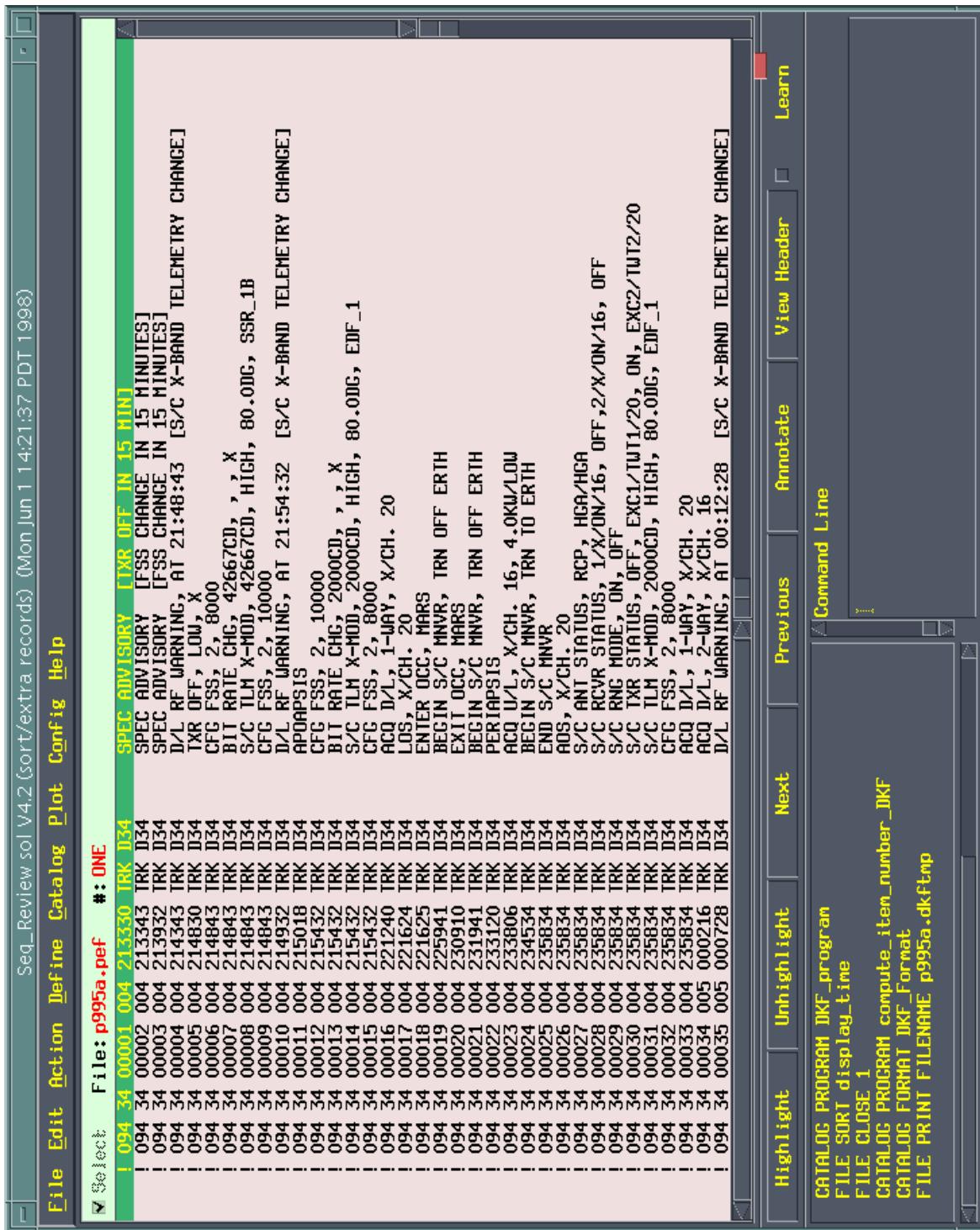


Figure A.6  
Option Used [-DKF]  
Example: **pef\_display -DKF p995a**